AFRL-RI-RS-TR-2008-293
Final Technical Report
November 2008

RAPID: RAPID PROTOTYPING IN DISTRIBUTED MISSION OPERATIONS (DMO) ENVIRONMENTS

Odyssey Research Associates, Inc.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2008-293 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

JAMES R. MILLIGAN
Work Unit Manager

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
NOV 2008	Final	Jun 05 – Sep 08

4. TITLE AND SUBTITLE

RAPID: RAPID PROTOTYPING IN DISTRIBUTED MISSION OPERATIONS (DMO) ENVIRONMENTS

5a. CONTRACT NUMBER
FA8750-08-D-0065/0003

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Ken McVearry, Ken Birman, Dan Freedman, Robert van Renesse, and Hakim Weatherspoon

5d. PROJECT NUMBER
558J

5e. TASK NUMBER
QB

5f. WORK UNIT NUMBER
03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Odyssey Research Associates, Inc.
33 Thornwood Drive, Suite 500
Ithaca, NY 14850-1280

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RISE
525 Brooks Rd.
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-293

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2008-0960

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goals of the RAPID study are to explore the limitations of current Distributed Mission Operations (DMO) systems, explore the emerging opportunities associated with Service Oriented Architecture (SOA), Global Information Grid (GIG) and Net-Centric Enterprise Services (NCES) technology trends, and to propose technical solutions and the trade-offs between them.

15. SUBJECT TERMS
DMO, SOA, GIG, NCES, HLA, DIS, distributed, mission, modeling, simulation, training, scenario exploration, federation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James R. Milligan
U	U	U	UU	41	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

Table of Figures

1. Introduction

1.1 Propose

This document assumes familiarity with the DoD Distributed Mission Operations (DMO) technology area, and notably with the DMO Focus Study conducted in early 2008. Our purpose here is to build upon the findings of the DMO Focus study, asking what technology options exist for advancing the state of the art in the DMO area to promote improved scalability, performance, and support to Wide Area Network (WAN) deployments.

1.2 What is DMO?

DMO is a technology base for creating distributed simulations for purposes of training, planning, wargaming and the like. The technology exists in an initial state as a federation of existing simulation solutions and systems, but there is a very ambitious roadmap for advancing DMO into the future. When we talk about DMO in this document, we are focused on two questions: How can DMO actually be implemented to achieve near-term goals? And how can DMO later leverage advanced technology developments to achieve its longer-term objectives?

DMO represents a compelling vision of a future in which DoD specialists can engage in training, scenario exploration and what-if analysis anytime, anywhere, pulling in a mixture of real-world data and highly realistic synthetic content.

To accomplish this goal, DMO needs to unify a diverse collection of modeling and simulation systems, implemented in distinct efforts by many teams worldwide, and operating in a variety of hardware and software configurations. While such steps are easy to summarize, accomplishing them poses huge challenges, and the goal of the RAPID study was to identify some of those challenges and to recommend technical steps that might alleviate them.

Before we talk about these challenges, we should comment on the commercial context. The DMO vision plays out against a backdrop consisting of a great variety of commercial gaming and Virtual Reality (VR) systems that are making money (DMO costs the DoD a lot of money and earns none). Google's Earth system may soon rival the DoD's own mapping resources, at least for those parts of the planet where hostilities are not underway. The associated flood of cash and the tens of thousands of developers associated with these trends permits the commercial world to advance by leaps and bounds. Thus, what DMO hopes to accomplish is already a daily reality for users of Second Life, the popular role-playing game, or World of Warcraft. On the commercial front one sees a tremendous diversity of hardware and software, still further standards, and a level of excitement the DoD could never create with its own in-house solutions. The obvious step is to somehow leverage all of this energy and technology, so that the future DMO would be a hypothetical fusion of a World-of-Warcraft-like technology, a Second-Life-like system, and yet drawing on real world databases that accurately model terrain, friendly and enemy forces, threat scenarios and the like. All of this would run on

the global DoD computer network, the Global Information Grid (GIG), and would need to respect the Web Services (WS) standards favored by that technology area.

In its initial steps towards a linkage of existing DoD simulation solutions, DMO is already encountering tough realities. First, existing DMO systems were never conceived with the goal of scaling to the degree needed by future deployments. Applications that were previously used with a few dozen components running in a dedicated LAN are suddenly being moved into settings where tens of thousands of components may interact over WAN distances, with firewalls, packet loss, high latencies, etc. Indeed, even linking the existing system together (without scaling them in any drastic way) is placing existing technology components under huge stress.

In the early days of computer science, a similar phenomenon bedeviled developers of applications to solve problems in settings such as factory floor process control. Over time, an entire theory of scalability emerged for the associated problems. Today, any computer scientist receives training in the analysis of algorithms with the goal of understanding their intrinsic scalability. We all know that sorting is an *n log n* problem, and that finding the shortest route that visits every node in a graph with labeled edges is an *NP complete* problem. Yet how many computer scientists think of *communications* problems in similar ways?

In fact, communications issues are every bit as complex and challenging as data structure issues seen in conventional non-distributed settings. One can design protocols that scale without limit in the numbers of participants, but this does not come for free; those protocols are often a bit sluggish. Some of the fastest ways of performing reliable event notification have *log* time complexity. Yet some of the most common ways of solving the problem have *linear* or even *quadratic* costs, as a function of the number of participants. And indeed there are many settings in which seemingly solid communications solutions, such as the popular publish-subscribe technology offerings (these match closely with one element of DMO), are observed to become chaotically unstable, thrashing and potentially shutting down entire data centers. Amazon, Yahoo and several other major data center operators have banned certain classes of communication products because of bad experiences of these sorts.

DMO, then, needs to develop and apply a methodology for predicting the scalability of the proposed large deployments and ensuring that they will be stable under stress. Failure to do so would be a recipe for future costly setbacks. Doing this today can lead DMO down a path to a future stable, scalable and high-value platform. These challenges parallel the ones confronted by companies such as Amazon, Google, and by the developers of games like Second Life and World of Warcraft. However, the companies mentioned have literally thrown away their code bases and rebuilt from scratch, often multiple times. They have the luxury of drawing on enormous financial revenue streams, and the ability to turn on a dime. In contrast, DoD needs to reuse existing solutions. Throwing out the current systems and rebuilding from scratch is just not an option. Accomplishing DMO goals while preserving the major platform legacy is thus a big part of our challenge.

Our discussion may sound as if it will argue for some form of theoretical research, but in fact this is not where it leads. In fact, our DMO study argues for selection of practical best-of-breed protocols, and their careful insertion into the DMO setting. But this raises a new question: DMO is a large legacy-based system. Inserting anything "new" into such a platform is potentially a tough proposition because changes may involve work by dozens of developers working independently on hundreds of components. Our study needs to ensure that its recommendations will not just be scalable, but also *practical*. We do this by focusing on classes of technologies that already exist in demonstrable forms, but also on mechanisms for inserting those into the DMO platforms that have been shown to require little or no change to the legacy solutions that run over them.

Although DMO is a moving target, our underlying assumption is that the trends discovered by the Focus Study will play out and permit existing DMO-based technologies to move towards the sort of unified system concept currently envisioned.

1.3 DMO Value Proposition

Our work builds upon a prior DMO "Focus Study". The DMO Focus Study looked closely at DMO as currently envisioned, and identified what might be called the DMO "value proposition". Understanding this background is key to understanding the ways that DMO might in the future be extended or modified.

We quote: "*DMO provides individual leaders, pilots, C4I operators, crews and teams, with realistic combined arms and /or coalition training and mission rehearsal opportunities. It provides the realistic voice and data communications and coordination in a near real time environment with other players in virtual battlespace. Often training is mission focused with external inputs and limited interaction with local operator/instructors. DMO provides the opportunity to train as we fight, allowing participants to train with the actual operators and units they would go to war with. Instead of training an F-15E crew on flying escort missions without the other elements of the package, DMO allows the crews to train/rehearse the escort missions with the pilots and crews that would fly the attack (F-16), jammers (F-16CJ), AWACS (E-3) and other aircraft associated with the mission. Often DMO provides the only opportunity to exercise the entire kill chain in a non-combat environment. DMO provides these types of meaningful training at a fraction of the cost of real operations. In addition to eliminating fuel, weapons and aircraft maintenance costs, DMO allows training on tasks so dangerous or costly that they are only done in combat.*" -- DMO Focus Study

These observations framed our own work. Rather than viewing DMO as an open slate, we adopted a narrower and more pragmatic approach: we view DMO as an attempt to accomplish the goals just articulated. Someday DMO will do more, but it is not our primary role to speculate about the long term. Rather, we want to ensure that DMO actually can achieve these initial objectives. And because of scalability challenges, that is not a small or simple matter. ***Our finding is that DMO is unlikely to scale adequately unless targeted investment is undertaken to address its communication needs in ways very remote from those currently being used.***

We note that the Live, Virtual, and Constructive (LVC) Architecture Roadmap (LVCAR) study, sponsored by the United States Joint Forces Command (USJFCOM) and which explored similar questions, reached very similar conclusions. Figure 1, extracted from that study, illustrates the way that the various DMO technologies "relate".

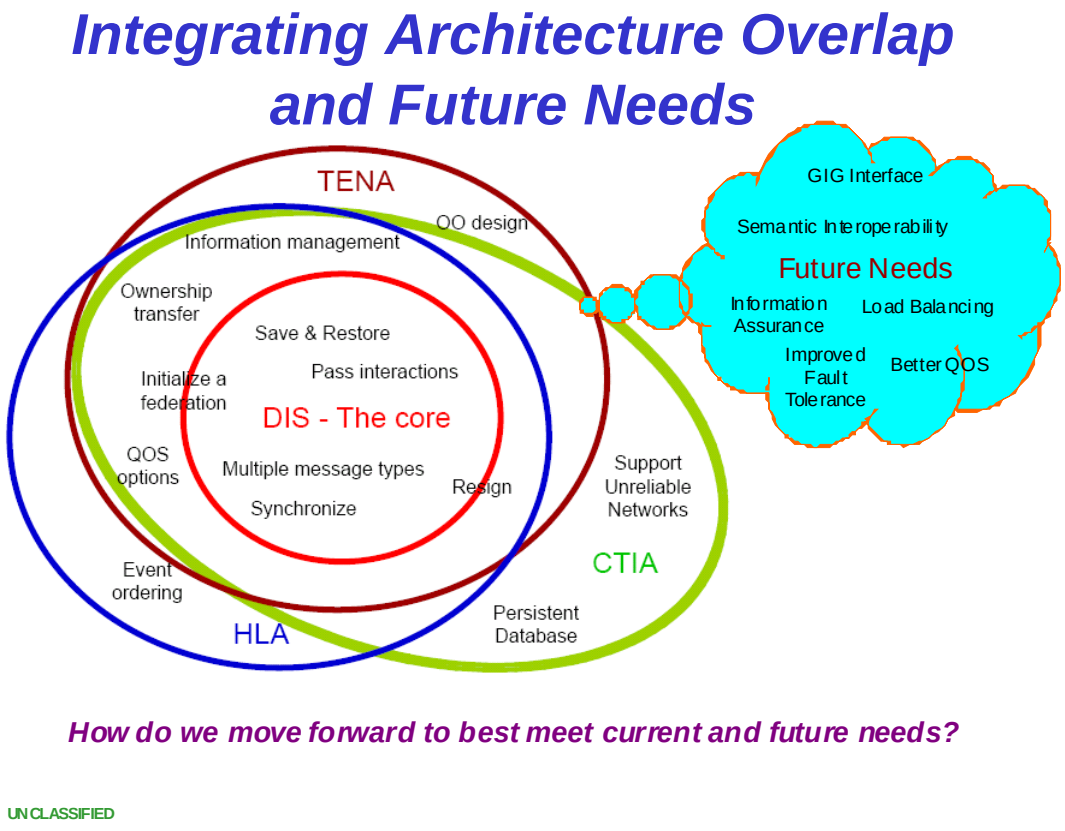

Figure 1: Overall DMO Vision

1.4 Limiting the Scope of the RAPID Study

RAPID was a short, focused study that began early in the summer of 2008 and finished in October of the same year. During this amount of time, it would have been tempting but unwise to pursue speculative tangents. It was necessary for us to severely limit the scope of the questions posed and answered.

With this in mind, and on the basis of the DMO value proposition just articulated, the RAPID study focused on the technology challenges of scaling DMO adequately to *exercise the entire kill chain in a non-combat environment.* We did not explore other potential configurations or uses of DMO, such as training pilots to deal with failures or completely out of the box situational challenges ("Chinese nanobot swarm attacks our troops while defending a military installation in Taiwan" kinds of scenarios). In particular, the RAPID study:

1. Assumed that we are talking mostly about *current* DMO technologies.
2. Assumed that the new system federates existing systems and simply tries to *run them on a larger scale than has previously been attempted.*

a. We assumed that this includes larger numbers of role-playing participants
b. We assumed that the computer network itself extends globally and that DMO systems are no longer being used primarily in dedicated data centers isolated from the GIG.
c. We assumed that the substantial legacy technology base must be preserved and leveraged and that rewriting existing solutions is not a realistic option at this stage.

3. We limited ourselves to asking questions about *communications technology.*

Given these narrower goals, the RAPID study sought to identify and document at least two technologies that could address the existing communications-imposed limitations of DMO Systems, with an emphasis on two core DMO proposals called HLA and DIS (discussed in more detail below), as used in wide-area GIG-style networks. We set out to compare and contrast these technology options both with one-another, with the existing DMO technology base, and with computer gaming protocols used in systems like the Second Life and World of Warcraft ones mentioned above.

As just noted, our first discovery is that ***as currently being implemented, DMO will fail to achieve its scalability objectives.*** We concluded this on the basis of a nearly trivial complexity-style scalability evaluation reflecting the underlying pattern of communication used in the most active DMO component technologies today (HLA and DIS). The core problem is that both presuppose styles of communication known to scale poorly. Indeed, both are on track to repeat the experience that led Amazon, Yahoo and others to "toss out" commercial products that they were using in-house, in similar ways!

We concluded that new communication options are going to be necessary. The RAPID study did not presuppose any particular technological solution, and indeed it was viewed as important to include both research and commercial products in the trade-off analysis. The team was asked to consider technical trends that might reshape the technology space over time, but also to limit itself to technologies that can potentially be "bolted on" to the existing DMO technology base without massive, disruptive change, thereby preserving DoD's legacy investment.

The RAPID study identified a number of candidate technological solutions. These were considered with respect to scalability, fault-tolerance, performance, security, and Quality of Service (QoS). Later in this document, we present evidence to support our findings, including summaries of research evaluations (published papers from the recent literature) that looked closely at many of the recommended technology options. Our discussion touches upon both new experimentation and benchmarking, and analyses of past experimentation and benchmarking results.

We are optimistic that if the best of breed solutions we identified are deployed into DMO, or new solutions are developed that use these existing options as blueprints but extend them without disrupting their scalability properties, the core DMO goals can be achieved.

When the RAPID study was initially proposed, it was anticipated that new experimental work might be needed to achieve the necessary confidence in the recommended steps. This was indeed necessary, but fortunately (in light of the short timeline of the study), the team was able to leverage ongoing experimentation funded by AFRL under a different

grant, the Castor effort, just ending a three-year effort. The experimental work undertaken by Castor as part of a sub-project called "Dr. Multicast", and in conjunction with the development of the Ricochet and Maelstrom protocols, turns out to be just what we needed for the RAPID recommendations relayed here. We note that the experiments in question took several years to design and carry out; no four-month study could have come close to doing anything equally ambitious.

1.5 Summary of Study Objectives and Summary of Findings

In summary:

1. The RAPID study focused on *existing goals* of the DMO technology space, with a goal of understanding *existing limitations and issues* and finding ways to overcome them.
2. We did this by understanding how DMO is intended to be used in support of its basic training mission, and deliberately steered away from hypothetical uses of DMO that might push well beyond the current envelope.
3. We placed enormous stress and importance on the exploitation of today's legacy solutions. Our understanding of the value of DMO revolves around the legacy of underlying technologies: simulations and associated support that already exists and that needs to be preserved as we move forward. Throwing large parts of the solution out and reimplementing them is not an option.
4. We focused on well-understood technologies from the commercial and research sectors, seeking to leverage them in ways that have predictable cost implications and should yield predictable performance and scalability benefits to the DMO base. The intent was to recommend practical, safe, short-term solutions.

Our study is detailed in the remainder of this document, but an executive summary would run as follows. First, we concluded that scalable communications will be a tough problem for the DMO architecture as currently implemented. The core problem is that protocols such as HLA and DIS (as well as others we mention below) will simply not scale in the intended manner. Worse still, the legacy systems are using HLA and DIS in ways that scale poorly. The combination is a recipe for failure.

Fortunately, we also see potential for a solution. Recent work has shown that one can often interpose a new communication solution into an existing system, by mimicking library routines that provide core functions to that existing system. HLA and DIS run over standard Internet protocols: unicast-oriented TCP and UDP broadcast. As a result, it is possible to talk about *transparently replacing* the TCP and UDP sessions with other protocols that have the same functionality (they deliver messages in the way that TCP or UDP broadcast would have), but work "better" in the sense of having better scalability, time-critical functionality, better tolerance of wide-area communication latencies, etc.

Our initial recommendation focuses on this category of intervention. While no panacea, it would seem that a highly targeted, narrow change to DMO systems could be fully or at least mostly transparent (no changes needed to the legacy code base), and that doing so would allow DMO to leverage a number of exciting new protocols that offer substantial benefits relative to the existing HLA and DIS implementations.

These new protocols include both off-the-shelf commercial products and solutions, and specialized solutions that would need to be implemented by DoD vendors, but for which public domain, open-source exemplars exist and hence where the risk associated with the new implementations would be minimal.

Although our study was firmly grounded in reality and much of what we recommend is solidly feasible in a short period of time, we are also well aware of longer term trends relevant to our area. With this in mind, we also looked a little further down the road and asked in what kinds of work DMO should invest in order to be in a position to repeat this sort of "short term, safe" intervention a few years from now.

Phrased differently, *speculative longer term research launched today will create tomorrow's low-risk, safe interventions for the future DMO of five or ten years hence.* We concluded that while research organizations such as NSF and DARPA are starting programs that will be relevant to DMO, no existing research organization is looking at the right mixture of questions, and hence the answers that those other programs arrive at will only represent partial stories for DMO. If DMO is to achieve that low-risk, high-confidence situation several years hence, DMO and AFRL need to tackle the higher risk and more speculative issues today, explore the most exciting potential questions, and in this manner start to elucidate the answers that will later be needed. Hoping that others will tackle the right questions poses substantial risk because if they fail to do so, DMO will not have the stories it needs to take its own next steps.

With this in mind, we also offer recommendations for longer term, higher-risk research. This set of recommendations revolves around what we call an "Event Net" architecture. The term is intended as a play on words: NSF is using the term "Data Net" to talk about scientific supercomputing centers that host huge amounts of data and are networked over wide area high-speed links to create repositories in support of scientific researchers who might work worldwide and who need to share their data and findings.

For us, Data Net research comes very close to what DMO will someday need; the connection is obvious. Yet DMO is not a federation of scientific researchers, working in a leisurely way and sharing work through published articles. DMO systems operate at high rates of events that report locations of assets and enemy forces, weather, weapons being fired, damage to physical infrastructure, etc. These streams of events are a pervasive and key component of the DMO environment, and they need to interplay in a seamless and scalable way with the DMO form of Data Nets. Thus our recommendation: DMO and AFRL should consider launching a research effort focused on Event Nets, consisting of Data Nets used in settings where high-rate event streams play as important a role as the underlying archival data stores. Knowledge gained in studying these emergent systems will later be the key enablers making possible advances in the DMO space itself, five or more years in the future, as our initial recommendations play out.

2. Current DMO standards and protocols examined

The RAPID study builds upon the initial DMO focus study, which started by looking at the current state of simulation architectures and protocols used for crew and operator training. There are two approaches that are the basis of most current DMO solutions, the Distributed Interactive Simulation (DIS) protocol and the High Level Architecture (HLA). In addition, the DMO Focus study that proceeded RAPID looked at the Test and Training Enabling Architecture (TENA) and the Distributed Mission Operations Network (DMON). TENA is a software implementation that is based on HLA. The DMON is a large DMO network that utilizes a modified version of DIS for its simulation backbone. Much of this section is based directly on the report of the DMO focus study.

2.1 Current Simulation Standards and Protocols

The world of simulation and VR modeling is very large, very intense, and as is typical of such communities, has a great diversity of competing and overlapping standards. To understand how this can occur, it may be helpful to just say a few words about what standards actually are, in computer networking settings.

Although the term *standard* sounds like a single universally adopted convention, in computer science the term long ago took on a different meaning: a standard, as used in the context of this document, is simply any formally documented set of rules, data representation conventions, protocols, or other agreed-upon application program interfaces. For example, there are several ways to represent a long integer in the memory of a computer (the byte order in memory can vary). Each of these ways was long ago standardized, hence there are several standards for representing an integer. Any given CPU adopts and implements some standard, and yet this does not ensure that when computers from different vendors are compared, they will turn out to be using the same conventions. Each is standard, and yet the standards differ.

Similarly, DMO is proposed as a federation architecture that interconnects simulation systems, which themselves are constructed using a set of overlapping standards. Unlike the example of integer representations, these standards do not actually "solve" the identical problem: they reflect systems built in very different ways and with very different underlying models of what kinds of data and entities are being modeled. Yet they also have similarities – they communicate over networks by exchange of messages. This ground truth explains what may seem to be a baffling situation, in which DMO appears to incorporate multiple standards for doing the same things in very different ways.

As noted earlier, it is not feasible to recommend that some single standard be adopted. Companies like the ones operating Second Life or World of Warcraft are fortunate: they can dictate a single standard and their developers will follow it; if the standard becomes problematic, they can recode their entire application. DoD lacks the resources to do anything similar. As a result, DMO will always be a world in which multiple standards overlap and co-exist. DMO is thus a *federation of simulation systems* and not a single new encompassing architecture. The main goal of DMO is just to create portals by which

these systems can cooperate and interoperate. Yet even that single step poses huge challenges, as noted earlier, and reiterated below.

2.2 Distributed Interactive Simulation Protocol

The DMO Focus Study concluded that although DMO will embody systems based on four primary standards, two of these are most heavily used and will continue to bear the brunt of the communication traffic. The first and most popular is a protocol called the Distributed Interactive Simulation Protocol, or DIS. The description that follows is taken directly from the DMO focus study.

"DIS provides a means to connect separate simulation applications using a defined set of messages and protocols. Through the utilization of DIS Protocol Data Units (PDU) the different applications can share entity data and interactions in addition to communications messages. The PDUs are divided into several families to include; Entity State, Fire, Detonate, Signal, Receiver, Transmitter, simulation control and others. These PDU messages are broadcast User Datagram Protocol (UDP) across the network and describe the entity location, moment, health and radio transmission and emissions.

DIS, which has been widely used in the Live Virtual and Constructive (LVC) simulations since the early 1990s, evolved from the US Defense Advanced Research Projects Agency (DARPA) sponsored SIMulator NETworking (SIMNET) project, which linked several US Army tank training systems to provide a distributed training environment in the 1980s. DIS is based on a set of agreed message definitions and enumerations that are defined in the 1278.1a-1998 IEEE standard. The Simulation Interoperability Standards Organization (SISO) sponsors changes to the standard for DIS. Recently changes that support directed energy weapons and transfer of entity control have been added to the standard.

In addition to the PDU message sets DIS defines a set of enumerations that identify many of the values in the PDUs. This includes things like the description of wave forms in Signal PDUs, entity type and appearance in Entity State PDUs, and weapon details in the Fire and Detonate PDUs. By defining this message set and enumerations the integration of multiple simulation applications is minimized." – DMO Focus Study.

For the purposes of the RAPID study, we focused one specific aspect of DIS: the use of UDP broadcast as a data dissemination technology. In a DIS application, every component receives every event notification.

Consider a system that slowly scales up. At first, perhaps we are simulating a single tank, then a small unit of tanks, studying possible tactics for attacking a command post armed with anti-tank weapons. It makes sense that every "event" be visible to every component of such a simulation: the troops operating each of the tanks need to see everything that is happening around them, and because the unit is physically close, everything is at least potentially visible to every "player".

But now suppose that we decide to scale to an engagement in which thousands of tanks will participate, thousands of aircraft are flying overhead, and tens of thousands of ground troops are participating. Clearly, a communications model in which every player sees every event in the entire system cannot be maintained. Either we will be forced to reduce the event rate as we add more participants (e.g., we hear about the state of an F16

not once every 100ms, but once every second, or once every minute), or the data rates that the participants will experience is certain to rise linearly in the number of players. No matter how fast the network and the machines may be, data will swamp them.

The situation is even worse than this makes it sound, because DIS was designed for use in low-latency (small delay), high speed, dedicated local area networks (LANs). One should imagine a single Ethernet with all the players sharing it, working at a small cluster of high-speed workstations. DIS entity state messages are broadcast based on a heartbeat. For example, ground entities that move slowly or not at all must update their state every 55 seconds, while an aircraft must update its state every 5 seconds as long as it is flying straight and level. If an entity is maneuvering, depending on the needed fidelity, the update rate can exceed 20 updates per second. Receiving simulations will drop any entities that fail to update inside a time window, which is a little more than twice the update rate. This allows for an entity state event to be lost without the entity itself vanishing from in the simulation. This approach also allows simulations that join late or drop-out and rejoin to receive an update on all entities in less than one minute.

When we spread our players worldwide, the UDP broadcast will need to be relayed over wide area links shared with other traffic, relayed through firewalls and travelling over long physical distances with substantial speed-of-light induced latency. Packet loss becomes an issue and reliability of the protocol will suffer. UDP broadcast itself is not widely supported in WAN networks because of concerns that it can disrupt router performance. The update event logic just described ensures that loads will grow dramatically: if n entities must broadcast their state to all $n-1$ other entities, the aggregated load on the network itself will rise as n^2. Under the reasonable assumption that entities are spread about in the network, most of this traffic will travel long distances and the core of the GIG will actually experience that n^2 load. Of course, the whole idea of UDP broadcast is that a single event can be communicated to many receivers in a single message, but a WAN network cannot support broadcast directly and routers must typically forward a message, copy by copy, on each outgoing link. Thus routers will actually experience this n^2 load.

This is so clearly a problem that we should perhaps say a bit more about it. The core problem relates to the way that the fastest modern routers operate. Normally a router that receives a single packet can send it in "one unit of time". With UDP broadcast, that one packet may require many send operations – k units of time, if there are k outgoing links. This violates the hardware model used by the highest speed routers, which consequently cannot support high data rates of UDP broadcast.

Thus, DIS is a problematic protocol for wide area deployments. Data rates will become extremely high. Routers will not be happy. And end-user systems will be overloaded by a flood of data, much of it relating to "entities" not directly visible to them.

One solves such problems at several levels. The DIS community is moving to introduce forms of hierarchical structure to their simulations: entities will only see events that are local to them. Filtering is being used heavily to ensure that unwanted data can be discarded as it arrives, offloading some of the work from the application protocol to its associated filter (which is often implemented as an attached co-processor on the network interface card). Military vendors are offering special WAN communication solutions

that bypass the standard military GIG and route the DIS traffic through a special purpose hardware network that can handle the loads. (This last option, of course, runs contrary to the GIG concept that a single military network will carry all non-sensitive traffic, with an additional single dark core network carrying classified traffic)

DIS poses additional problems. The DIS message format includes an "entity identifier" that uniquely describes the object that generated an event. In theory, every round of ammunition has its own unique identifier. According to the baseline DMO "focus" study used as background to the present report, when using the current representation, the theoretical maximum number of entities generated at any one site in an exercise is limited to ~ 4.29 billion entities, a number stemming from the size of the numeric field used to represent the identifiers. While this is a large number, it may be easier to exhaust than one might expect: to simplify the generation of entity identifiers, DIS applications apparently use numbering schemes that might not make use of every possible value. As a result, one can easily imagine scenarios in which DMO succeeds in enabling much larger simulations, only to encounter DIS scalability limitations. Simply increasing the DIS identifier field size is one obvious option, but at the cost of requiring modifications to existing legacy applications. It thus makes sense to ask if there might not be options that could "evade" the limitations but in a manner transparent to existing applications.

Such limits pose a concern reminiscent of the concerns about IP addresses in IPv4, the Internet standard which has a similar limit and actually encountered it more than a decade ago, forcing the introduction of so-called *network address translation* (NAT) units. It seems clear that for global deployments of simulations with large numbers of participants some similar form of DIS entity identifier translation technology will eventually be needed. Our initial belief is that the NAT model could be adapted for use with DIS, and is largely transparent: applications are mostly unaware that NAT boxes are present.

We have commented that DIS has some obvious problems when moved into WAN environments with large numbers of entities using the protocol. But DIS also has several limitations as a result of the way most simulation applications handle DIS event reception and processing. We mentioned one of these above: the need for every event to reach every entity, even though most entities are only "interested in" a tiny subset of the DIS events. The need to filter and discard unwanted DIS events is a huge limitation of the protocol. Worse still, in large complex exercises, one can easily run into a situation in which a "new" weapon system or vehicle broadcasts a type of DIS event never previously seen. Some applications might mistakenly attempt to process those events (or might believe themselves to be required to do so), and could then crash over the unknown event type as they try to make sense of it.

2.3 High Level Architecture (HLA)

Like DIS, HLA is a software architecture for creating computer simulations out of component simulations. However, where DIS is based on a model in which every simulated entity broadcasts events that every other entity will see, HLA uses a more structured approach in which each event is associated with some class of objects and only relayed to entities interested in that object. For purposes of background, we start by including overview material drawn from the DMO "focus" study; readers familiar with

that study may wish to skip ahead to the non-italicized content, which is new to this report.

From the DMO focus study: *HLA provides a general framework within which simulation developers can structure and describe their simulation applications. HLA was developed by the Defense Modeling and Simulation Office (DMSO) as a replacement for DIS and other simulation protocols in starting in mid 1990s. There are a number of HLA implementations by multiple vendors, called Run Time Infrastructures (RTIs). The underlying concept clearly borrows from the well known CORBA standard for interconnecting object-oriented applications and systems.*

HLA has a flexibility not found in DIS, with the ability to define the models that represent the entities and interactions in the simulation. Anything can be communicated using HLA, as long as it is defined in the Federation Object Model (FOM), the Simulation Object Model (SOM) for its originating simulation, and the Management Object Model (MOM). The FOM defines the attributes of all objects in a Federation to include entities and simulation messages or interactions. The SOM defines the attributes of a simulation entity (as defined in the FOM) that a particular Federate will update and reflect (i.e. publish) so that other Federates may be informed about changes to the entity (i.e. subscribe). The FOM for a Federation can be considered to be the union of the public parts of the SOMs for all of the participating Federates. The MOM defines the interactions that are used to manage the Federation, addressing things like, starting / stopping the federation and time management.

The flexibility that HLA provides comes with a cost of limiting interoperability. In addition to the requirement that each Federate run the same version of RTI, each must also run and support the same FED and RID files. Whereas DIS simulations should work together with little coordination, HLA requires detailed coordination and may require some code compilation before different HLA simulations will work together successfully.

For purposes of the RAPID study, the question to ask about HLA concerns its scalability. HLA employs a form of publish-subscribe model, in which an entity that wishes to monitor some other entity "subscribes" to it. In most HLA RTIs, this is done by making a TCP connection to the publishing entity and registering some form of interest. As the monitored entity changes state, it sends event messages in a form specific to the kind of object being modeled to its subscribers, which are expected to interpret those appropriately.

Scalability of publish-subscribe systems is a well known problem, particularly in WAN settings but evident even if one scales up a data center using the publish-subscribe model in a local area network. There are a number of issues, and all of them manifest in sharply *decreasing performance* as a function of the number of subscribers, or of increased link Round Trip Time (RTT), or even modest link data loss:

1. For reliability purposes, protocols such as TCP buffer copies of each outgoing message until it has been acknowledged. The amount of space required quickly becomes enormous when modeling complex objects at high data rates. Yet the data is highly redundant: any single event may have been sent to hundreds or even thousands of subscribers!

2. In WAN settings, network latency can be very high simply because of long physical round-trip delays (even at the speed of light, cross-country latencies are 50ms when using standard optical routers and links). TCP is extremely latency sensitive and performance is poor in such settings unless the sender buffer is made extremely large, roughly as a linear function of the latency. Moreover, if any packet loss occurs, TCP will experience noticeable pauses linked to the RTT because lost data is always recovered from the sender. Thus, an application that needs to stream data at high rates will see increasingly disruptive hiccups when running on a WAN link, even with tiny loss rates. With loss rates as low as $1/10^{th}$ of 1%, TCP performance drops to near zero on links with 100ms RTT latency.
3. The act of sending an event becomes very costly. Whereas a DIS event requires a single UDP broadcast to be sent (albeit unreliably), an HLA publication will trigger one send operation per client. The operating system itself is poorly equipped to support applications with more than a handful of TCP connections, yet HLA could easily require hundreds or thousands of connections in large configurations of the sort DMO seeks to support.
4. Latency skew becomes a noticeable problem: the *n'th* copy of an event may reach its subscriber so long after the *1'st* copy did that fairness becomes a substantial issue. For example, imagine an application in which a dozen combatants witness the same enemy threat, but in which the notifications spread over a period of a few seconds: perhaps, the threat becomes visible to user A at time 10:00:00 and to user B at time 10:00:03. User A might achieve a sharply higher kill rate simply by virtue of the spread in notification times, potentially leading to rewards, while user B might be penalized. A "fair" system would provide the same data to all receivers at the same time, creating a flat playing field in which the best performance reflects the actual user's behavior, not an artifact of the communication system. This concern will lead us to recommend experiments with time-critical multicast protocols that might enhance DIS performance in a scalable manner.
5. Weakest link phenomenon an issue: a single slow receiver can be extremely disruptive. The channel to it will fill up, and this eventually causes the sender to block when it tries to send events. In DIS that receiver would just miss packets because its UDP socket would overflow, so in DIS, the receiver falls behind, but in HLA, the sender ends up pausing to wait for the slowest receiver to catch up.

In industry, such problems typically motivate a move from TCP towards UDP multicast, where feasible and indeed, some HLA RTIs may be using UDP multicast (not broadcast – a multicast is selective and only goes to the appropriate receivers). Such an RTI could run over COTS solutions from companies such as TIBCO or IBM. However, these UDP multicast-based products can sometimes become unstable under stress of large deployments, high data rates, and users that come and go, which is why Amazon and Yahoo "threw out" products built in this manner.

A frustration is that the products in question normally work quite well, but it can be hard to predict which configurations will cause instability. Later, we will see that a new generation of protocols may allow the selective use of multicast to speed up HLA RTIs, and may also offer a remedy in the event that a product using multicast unexpectedly becomes unstable.

2.4 Test and Training Enabling Architecture (TENA)

There are several approaches to simulation that have been implemented to enhance training. This includes the Test and Training Enabling Architecture (TENA). Again, we use italics to denote material drawn directly from the DMO Focus study, and included purely to ensure that the reader of the current document has the needed information to fully appreciate the DMO challenge.

From the DMO Focus study: *TENA serves as the DoD integration architecture to enable interoperability among range systems, facilities, simulations, and C4ISR systems. It uses a HLA like protocol combined with TENA middleware and TENA objects. TENA is as much a software design architecture as a simulation architecture.*

TENA uses the term logical range to refer to a suite of TENA resources, sharing a common object model, that work together for a given range or ranges. TENA resources can be real radars, and aircraft at a test or exercise range, data loggers or simulated equipment. Additionally by abstracting things like coordinate systems TENA allows non-compatible equipment to function together.

The TENA architecture (Figure 2) provides application developers and integrators the ability to link and compile with the TENA middleware and TENA object model code when building TENA compliant applications. Additionally this allows them to take advantage of the defined and maintained simulation interfaces. The middleware provides translation between different coordinate systems in addition to other services. TENA provides web-based C++ code generation. This allows bug fixes and code updates without having to distribute the changes to the users. The auto code generation also allows the use of standard, validated algorithms (such as coordinate translations or unit conversions) that are embedded in TENA, relieving the burden of managing and performing translations from the software applications.

The core of TENA is a common infrastructure, shown below. This includes the TENA Middleware, the TENA Repository, and the TENA Logical Range Data Archive. TENA also identifies a number of tools and utilities. The TENA object model encodes all of the information that is transferred between systems during a range event. It is the common language with which all TENA applications communicate.

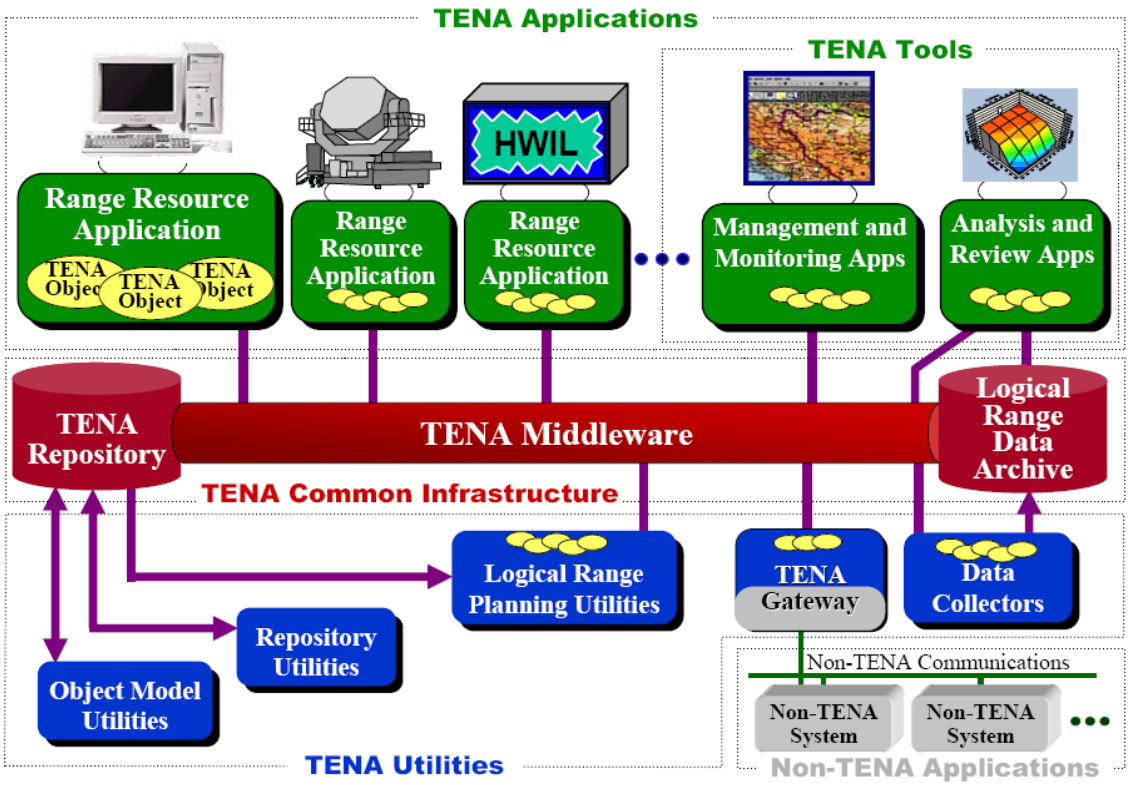

Figure 2: TENA architecture

From the perspective of the RAPID study, we can separate the TENA design perspective just articulated from the underlying communication questions, which represent the primary issue to which our attention is addressed. TENA communication occurs through the middleware component, which defines a high-performance, real-time, low-latency communication infrastructure. Superficially, this infrastructure is rather similar to HLA. The same scalability issues raised in the previous section thus arise again here, in connection with TENA applications that scale into very large deployments. The core problem is that the goals of high performance, time-critical or real-time behavior, and low latency are somewhat at odds with scalability, at least in the absence of a clear technical reason for believing that the desired properties will be achieved. Thus, a TENA system that demonstrates snappy interactive performance in a scenario with a few dozen participants may bog down as data distribution costs become an issue (recall the "fairness" problems mentioned before).

The key insight here is that just as one talks about complexity costs for data structures and algorithms – the *n log(n)* complexity of sorting, for example, or the *log (n)* cost of searching, there are also complexity costs associated with distribution of data in complex systems such as TENA and HLA. Merely stating that an architecture is focused on real-time behavior, oddly, can make such problems *worse* rather than better, since it implies that developers can safely assume that this property holds. If, as the number of users increases, the real-time properties degrade sharply, applications insensitive to real-time may merely seem sluggish or "unfair". In contrast, applications designed to assume a real-time substrate could easily crash.

Nonetheless, the core questions to ask about TENA communication are very much the same ones that arose when we examined HLA. These can be asked in two ways. On the positive side, the issue is to understand the best possible means of achieving the desired real-time behavior. Given state-of-the-art protocol support, how well could TENA do? (Quite well, in our view). But the negative question would ask how the existing commercial TENA platforms actually implement the needed communication layer. Here, the story seems likely to vary, but not likely to depart dramatically from the way that other DoD systems perform real-time communication: using technologies such as the OMG-standard Data Distribution System (DDS). These are indeed capable of blindingly fast message delivery when a sender transmits to a single receiver, but the situation is sharply different as the number of receivers scales up: the real-time guarantees are lost and performance degrades linearly in the number of receivers, because sending one message may require sending *n* network-level packets, to the *n* receivers.

Thus, as we look at support for scalability in TENA, we really need to distinguish two issues. One involves ensuring that in large-scale settings, TENA uses the best possible solutions. And the other is to ensure that the TENA guarantees themselves evolve to communicate the real-time costs of scalability to TENA developers. For this RAPID study, we focus on the former problem below, and leave the latter to others more oriented towards TENA per-se and to its evolving standards and best-practice documents.

TENA Gateways

So-called *gateway applications* allow the integration of TENA applications with non-TENA resources. Gateways communicate with both a TENA logical range (via the TENA Middleware) and another set of applications using some other protocol. Gateways contain algorithms that specify how to subscribe intelligently to only the information needed by that particular gateway. In larger systems, gateways might have to be federated with one another, working together to balance processing or network load. The TENA-HLA gateway is an important gateway which allows easy integration of TENA resources with modeling and simulation systems.

In our RAPID study, we concluded that gateway systems may offer an eventual opportunity for intervention when TENA platforms are networked to one-another, or to other non-TENA simulations. However, we did not have an opportunity to delve deeply into this issue.

2.5 Distributed Mission Operations Network (DMON)

From the DMO Focus Study: *The DMON is a persistent training simulation network that provides an on-demand virtual/constructive environment for mission training and rehearsal of air crew and C4I operators. The DMON provides and maintains the network backbone and a connection portal at each location. Currently there are almost 60 nodes at over 30 different locations around the world that are either active or in the process of joining the DMON supporting training for A-10s, Apaches, F-16C, F-16CJ, F-15C/Es, B-1s, B-52s, JSTARs, AWACS, CAOC at Nellis, DMOC and DTOC.*

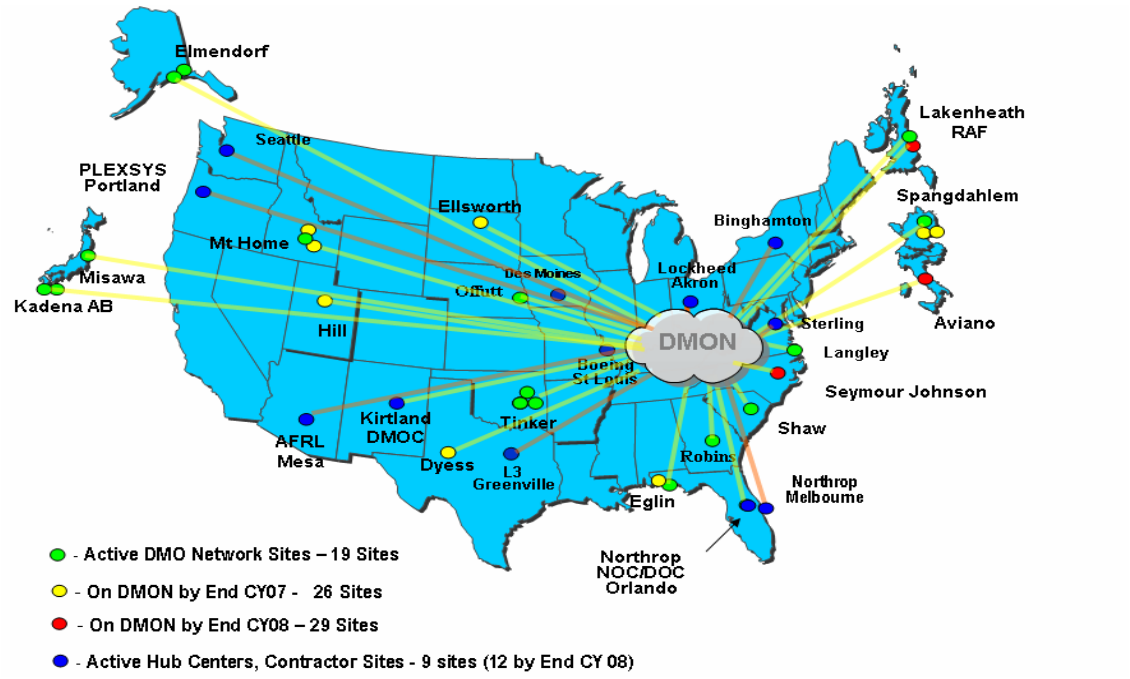

Figure 3: DMON deployment

The DMON implements a Virtual Private Network (VPN) that provides connectivity between the sites as well as the means for continuous monitoring and control of the DMON from the Network Operations Center (NOC). The NOC is located at the Northrop Grumman Mission Systems (NGMS) facility in Orlando, FL. The NOC provides resources for DMO Operations & Integration (O&I) personnel to conduct day-to-day operations of the help desk, and to administer, monitor, and maintain the private network used for training exercises. Figure 3 illustrates this communications graph.

A critical component of the DMON is the Portal which supports the DMO training system by isolating one Mission Training Center (MTC) implementation from another. It also facilitates communication among MTCs which implement different simulation protocols (e.g. HLA, DIS) across the DMON. Additional benefits provided by the Portal include the traffic management between MTCs, filtering invalid or unnecessary data produced by an MTC, routing traffic to MTCs based on simulation data values or needs, and a common user interface for MTCs to manage or view the status of a Combat Air Forces (CAF) DMO event. Portal functionality includes a state database, Dead Reckoning (DR), support for NATO EX (NEX) simulation protocol, support for multiple Local Area Network (LAN) endpoints supporting similar or disparate protocols (DIS, HLA, NATO EX, TENA), and Data Streams over the Wide Area Network.

DMON is currently quite "CAP-centric". The authors of the DMO Focus Study know of no effort to support MAP or other non-CAP capabilities in DMON settings. However, DMON does provide connectivity for AWACS and JSTARS in support of CAF missions.

Figure 4 illustrates the high-level components of the Portal architecture. With outbound traffic, an MTC LAN endpoint receives all network traffic from the simulator and is translated and forwarded to the Stream Manager. The Stream Manager receives

standardized data packets and determines which Data Stream(s) each packet must be associated with. Packets are then duplicated and passed along to the Distributor for distribution among the remote MTC sites. In the WAN Controller the out bound traffic can be filtered and the valid traffic is translated into Portal Speak (the protocol transmitted between Portals and representing the CAF DMO protocol standards) and sent to the remote portals.

With inbound simulation traffic from remote portals, the WAN Controller receives Portal Speak packets from all remote Portals listed in the configuration file. These packets are passed on to the Stream Manager to be routed to the proper LAN endpoint(s) based on the stream subscriptions. Once at the LAN endpoint, the packets are translated into the native MTC simulation protocol and sent to the network of the local MTC.

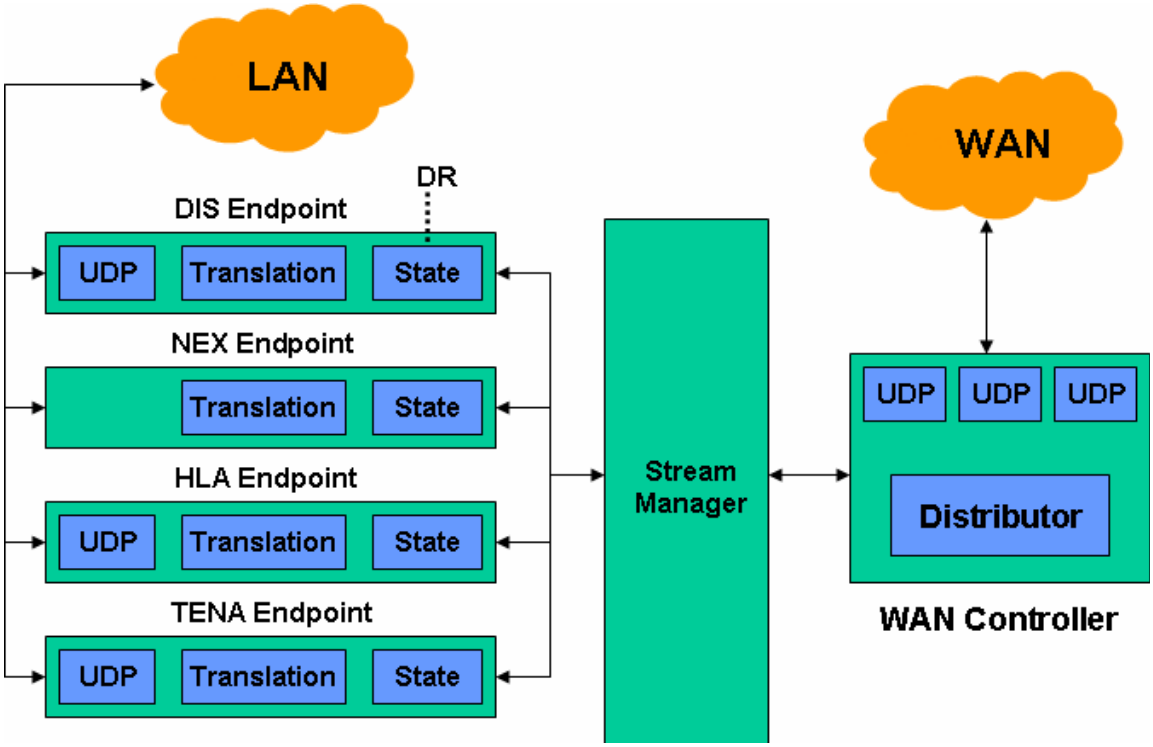

Figure 4: DMON used as a gateway to link DMO components

The RAPID study views the DMON effort as a potentially useful technology because of its role as kind of unifying gateway between DMO component systems. As noted earlier, such gateways represent a potential point for intervention at which the opportunity exists, at least theoretically, to exploit new kinds of protocols or new technologies capable of overcoming possible scalability bottlenecks.

3. DMO Communication Challenges

3.1 Summary of Problems Identified

We now summarize the basic findings of our study. As discussed in the review of DMO technologies, there is no single story that completely covers the challenges DMO will confront. Instead, we see a collection of stories, each raising significant challenges of its

own. Section references are included to assist the reader in identifying the relevant technology discussion, above

- [Sec. 2.2] DIS applications employ what can only be called an "inherently non-scalable" protocol, because of DIS's UDP broadcast architecture, originally intended for use in small, dedicated, LAN settings. The underlying approach (broadcast) is often translated in hardware to point-to-point messages. Hence, if n entities use broadcast, this imposes a potentially n^2 load on the router hardware. It follows that load on the central network routers could increase as the square of the number of entities using DIS. Even if hardware multicast is available (and it often is not), the load rises linearly in the number of entities. This is disturbing because even a linear load is potentially prohibitive. Today, DIS simulations would rarely have more than thousands of entities, but the day is coming when they may have millions. Thus participating applications could see orders of magnitude increases in inbound traffic. Even if we plan to filter and discard unwanted traffic, in this model the traffic must be delivered before the filtering action can occur. Thus, merely increasing the number of DIS entities potentially forces us to increase the capacity of the network links attached to each machine!
- As we saw in Sec. 2.2, DIS uses broadcast in a completely non-selective manner. Every entity receives every event from every other entity. Ironically, most entities are uninterested in most event messages, so after going to the trouble of delivering the data, the vast majority is filtered on reception and discarded.
- DIS does not require reliability, but in fact is designed for a setting in which most messages will usually be reliably delivered. If used in settings where UDP packets are often dropped, the typical DIS application could be destabilized.
- As noted in 2.2, DIS also faces a potential exhaustion of the entity identification space. Earlier, we hinted at the idea of using a scheme similar to that used in network address translator boxes to virtualize and enlarge the identifier space. But this is just an idea – today, any DIS application, no matter how large, must somehow make due with a finite space of entity identifiers.
- In 2.3, we saw that HLA employs a human-intensive federation approach, in which each entity is hand-designed and has its own, specialized, per-object state. Updates are sent (using an efficient representation), but as a consequence, every subscriber must be able to make sense of every update. Each application added to an HLA system potentially requires every existing HLA application with which it interacts to be upgraded to support it. Modern programming languages sometimes support dynamically downloaded and linked code that can extend a running system without forcing recompilation. However, most existing HLA applications predate these languages and runtime platforms, and hence scaling HLA to accommodate more and more kinds of entities is a potentially programmer-intensive activity.
- Also in 2.3, we saw that the underlying HLA publish-subscribe model is typically implemented using point-to-point TCP connections, a non-scalable architecture that introduces a number of issues. We elaborated many of these in that section, such as the difficulty of sending updates to large numbers of receivers. Mechanisms that work extremely well when only one application is watching a particular object may become very sluggish if thousands suddenly subscribe;

indeed, there are many publish-subscribe systems that simply cannot be used with vast numbers of receivers or in extremely large deployments: they collapse by slowing down linearly in the number of receivers watching any given object.

- Our comments about these limitations of publish-subscribe are not true for all possible HLA RTIs. Some HLA RTIs potentially run over UDP multicast products such as the TIBCO message bus or the IBM DCS platform. These, however, have their own stability problems and are known to exhibit problematic behavior when run in large-scale demanding settings under high data rates.
- TENA and DMON are intriguing technologies that clearly merit further study. Our preliminary findings, discussed in 2.4, focused on the concern that TENA applications often have real-time properties that may not hold as a system scales up. To the extent that these properties are *expected* to hold, those applications might fail. But the many concerns articulated about HLA seem to be applicable to TENA systems as well. We note that in 2.4 and 2.5, opportunities to potentially intervene in TENA and DMON systems, modifying them "transparently", appear to be present. This encourages us that with further study, TENA and DMON may be attractive targets for the kinds of technology innovations proposed below.

Although our overall conclusions may seem pessimistic, this is not actually the case. In fact we believe that DMO is unlikely to succeed if existing solutions are simply glued together and dropped, without modification, into WAN environments with vastly increased numbers of users. Indeed, the DMO Focus Study pointed to evidence that the technology base is already reaching its limits to scale beyond current WAN deployments and exercises, which should be considered modest when compared to the more futuristic vision of DMO. However we are quite optimistic about the prospect for overcoming many of the issues raised, and perhaps even catalyzing a major new advance in the capabilities of DoD simulation platforms by doing so.

3.2 Summary of Constraints

As noted in the introduction, we consider certain constraints to be fundamental to the RAPID study:

- DMO is a federation of large existing systems that reflect decades of investment, and jointly consist of a high-value legacy. Preservation of this legacy is central.
- In some cases, source code is not readily available for DMO components, or not easily modified (changed source code may violate support agreements).
- Because DMO is a community of communities, interventions that require broad consensus or agreement are probably not practical.

3.3 Relevant Technologies

Perhaps the most important contribution of our study was to identify candidate technologies that represent promising options for moving the DMO community forward. A second key issue is to lay out a plan of action for actually exploiting these options in real DMO systems. We address the first of these questions here, and the second in Sections 4 and 5.

3.3.1 Commercial Trends: Event Notification Technologies

After a long period of intense commercial competition during the 1980's and early 1990's, improvements in the raw speed of computer networks reduced the level of interest in scalable high-performance event notification solutions. As a consequence, a hiatus occurred, with a small set of majority products emerging and dominating what became a stable and rather mature market. Major products include:

- The TIBCO communication bus product, TIB
- IBM's Distributed Communication System (DCS), supported by a variety of hardware and software tools
- Microsoft's Event Notification technology for .NET

There are also a great number of smaller companies with niche products in the space.

While it is dangerous to generalize, and much information is anecdotal, the authors of this study happen to have had very good discussions with some of the companies mentioned above, and with some of the very largest commercial users of their products. In what follows, we repeat what we have been told, and yet must express a caution and some frustration at the lack of solid evidence for or against the whispered concerns. The original plan of the RAPID study was to try and do some experiments on major products, but, as we now explain, that proved to be impractical.

There are two core reasons for the lack of solid empirical research on the scalability and stability of major commercial products in this space.

First, one of the early major vendors was for a period so dominant as to be in a position to make "unusual" requests in its commercial contracts, which are themselves subject to non-disclosure limitations. We are informed by our contacts that among these requests, customers were required to accept strict limitations that expose the customer to legal liability if they benchmark and publish on performance issues. We note that even confirmation of this contractual language has been impossible for the authors of the RAPID study. We find ourselves in the frustrating position of having been told of something that may or may not be true. It is, however, certainly possible that companies in the space have taken steps to prevent their users from publishing performance studies.

A second problem is that the cost of setting up a large and demanding testbed, purchasing and deploying the products in question, and then testing them under laboratory conditions, would be prohibitive. Thus, while it would have been appealing to set up experiments of our own, doing so would require access to large numbers of machines, licenses for the various products on those machines, and a whole testing methodology. We determined that the costs associated with such a process would be staggering.

What this adds up to, however, is that we are in an unusual situation of being able to repeat authoritative concerns about the products, and yet not able to cite reputable scientific studies that document the underlying issues. Those who spoke to us did so on conditions of anonymity, but we note that they included very high-placed executives in some of the largest and most technically sophisticated companies in the business.

First, *we have been warned that in large deployments, event notification technologies suffer from pervasive performance issues*. If they work at all, they slow down as the deployment scales up. Earlier, we pointed to some of the underlying issues: the basic technologies available (TCP, point-to-point UDP, UDP broadcast and UDP multicast) all face tough challenges when scaled up and subjected to heavy loads. In some cases, we are able to document the problems (we will discuss this under the research findings sections that follow). But in no case were we able to take specific products and pin down specific scenarios in which those products demonstrably fail, as documented by public materials.

Second, there are issues associated with underlying reliability protocols. When permitted to operate unreliably, many of the same technologies just mentioned exhibit a degree of message loss that rises as the size of the deployment and the loads are increased. Yet *when reliability mechanisms are introduced, large scale deployments often trigger communication layer instability*. For example, one big dot-com retailer operates data centers with tens of thousands of machines in each, and for a period of time made heavy use of the TIBCO communication bus, a publish-subscribe product, in its reliable mode. As the company scaled to larger and larger deployments and higher and higher data rates, they began to experience data-center wide "blackouts" in which the TIBCO product was found to be saturating their gigabit LAN network with staggering rates of retransmission requests and retransmissions of data, and no good traffic was able to enter from any source at all. Indeed, applications that made no use of TIBCO at all were completely shut down, so severe was the overload condition.

In effect, TIBCO worked well for this company until a certain threshold of load and scale was reached, but the product then completely destabilized, shutting down the entire data center in a manner that disrupted all applications using it! One can easily imagine the consequences for the DoD if a DMO application were to have a similar impact on the GIG.

In the particular case, through interviews with individuals involved, we were able to pin down the cause of the problem, which can be summarized as follows. The product, TIB, used UDP multicast to send packets, and used point-to-point UDP messages to recover lost data. In patterns where there were many receivers and a few senders, it was possible for TIB to operate very efficiently. But as the system scaled, UDP became somewhat unreliable, and the typical multicast was dropped by some of its receivers. This caused a sudden spike in retransmission requests, and the extra load seems to have increased the unreliability of the UDP transport. In a cyclic meltdown, load spiked and reliability plummeted, with the combined loads from all of this traffic exceeding the very high bandwidth of the company's cutting-edge data center LAN.

Our study group then spoke to a well known major company that has another of the most popular products in the space, and was surprised when that company confirmed, under condition of non-disclosure, that their product experiences similar instabilities on large

deployments under heavy load and stress. Thus, we find that two of the most important products share a problem.

As we look to DMO, this is disturbing, because there is no obvious reason that DMO should be able to escape the same issues that these major vendors are struggling to overcome. A naïve DMO implementation would probably repeat the experience of the dot-com company we interviewed: it would perform well in small lab deployments, but would collapse as real use forced it to scale into much larger and more demanding settings.

We contacted companies to learn about in-house work to build better and more scalable event distribution solutions, and found that at IBM there is a significant effort underway. IBM has assembled a major R&D group that is tasked to build Cloud Computing solutions that would include high-speed, stable, scalable, event distribution solutions. A discussion of Cloud Computing appears in the next subsection.

We also determined that Red Hat is planning to create a scalable time-critical platform as a part of a planned new Linux release, targeted for 2010. (See below in connection with the Ricochet protocol).

We conclude that at least some of the major technology vendors do have plans to develop and market better event notification solutions. We thus see a danger but also a possible remedy. The danger is that as things stand, DMO is very unlikely to have a satisfactory experience as DIS and HLA are scaled up. The remedy is that products from IBM, Red Hat and perhaps other companies as well, will eventually be marketed in this space.

We should note that Cornell researchers are particularly well known for their work on this topic, and indeed that Cornell's Quicksilver system, developed with AFRL funding, may be the world's most scalable and stable event notification platform to date. The work underlying Quicksilver is readily accessible to vendors interested in learning how Cornell accomplished this through papers, and Quicksilver itself is available for download.

We conclude that the problem can be solved, and is being solved, but also that existing commercial platforms embody serious performance and scalability constraints. Failure to respect their limits can result in destabilized and disruptive behavior even at the scale of entire enterprise data centers. Moreover, given that our dialog was with world experts operating some of the most successful enterprises of this sort that exist, it is quite unlikely that DMO will achieve a better result unless it leverages the best of breed options. The underlying issue, as was seen earlier, can be understood to be a form of a little recognized "complexity" issue similar to the issues that make an $n \log n$ sorting algorithm superior to an n^2 sorting algorithm.

3.3.2 Cloud Computing

A particularly exciting commercial trend involves the explosive growth of so-called Cloud Computing platforms, which are basically large data centers architected to host a diversity of applications that might traditionally have run on client computing systems.

For example, Google offers e-mail, word processing, presentation, and other Cloud-hosted solutions, Amazon has storage for potentially enormous amounts of end-user data, etc. A secondary Cloud Computing trend involves virtualization: there is growing interest in taking PC images, virtualizing them, and running the results on Cloud platforms. Doing so has the benefit that if only one of, say, 50 machines is active, the Cloud essentially needs $1/50^{th}$ of the resources to provide the same snappy response the user expects. One can also slash management and other costs.

Within the Air Force, Cloud Computing is sometimes referred to as *consolidation.* This term is a reference to the property just mentioned: potentially, a large number of virtual machines can share a smaller number of powerful servers in a data center environment. The benefit of consolidation is (potentially) reduced cost of management and power efficiencies. Moreover, Air Force personnel, running on the edge, can potentially use web interfaces to obtain anytime, anywhere access to the resources in the Cloud. Yet one can also see the obvious complexities of such a vision, at least for the Air Force, where poor connectivity is unfortunately quite common, latencies can be huge, and mobility is the rule. None of these properties holds in the most successful Cloud ventures.

The RAPID study was not tasked to look at Cloud trends in general, nor did we have time to explore the full ramifications of Cloud Computing for DMO. Clearly, these may be dramatic: one could potentially host simulations in Cloud settings, and doing so might ameliorate some of the scalability and latency issues discussed earlier. On the other hand, doing so might break real-time assumptions embedded into TENA applications, and the high latency from some DMO users to a Cloud hosted in, say, Washington would create issues not seen today.

What we do wish to note is that in building Cloud platforms, researchers at companies such as Microsoft, eBay and Amazon are discovering powerful rules of thumb that can stand as guidance to researchers interested in future DMO scalability. For example, in a talk on the scalability of the eBay platform (see the online slide decks at www.cs.cornell.edu/projects/**ladis2008**/), eBay's Randy Shoup laid out a set of principles that dominate the eBay Cloud architecture. That architecture shares a great deal of commonality with many of the DMO objectives. Yet whereas DMO aims at providing a highly consistent view of the simulation state, eBay turns out to view consistency as the enemy of scalability. As much as possible, eBay allows individual users and systems to see events in different orders, perhaps even not to see the same events at all, and to reconcile state later, not immediately. They argue for a model called idempotence, in which actions can be applied again and again to the system state with only the first instance leaving any form of long-term effect and the repetitions, if any, simply returning the same result as the first version did. They partition workload and then, within each partition, partition the work even further.

Without wanting to prejudge a complex question, we see enormous opportunity for DMO in these kinds of architectural insights. The simple reality is that systems like eBay did not scale to run on hundreds of thousands of servers painlessly, and DMO can avoid that pain by learning from these prior examples. Many of the insights that companies like

eBay, Microsoft and Google are communicating through their papers point to opportunities for DMO, if cleverly applied. And the consistency point just mentioned certainly has implications for DMO communication. After all, DIS, HLA, TENA and DMON all provide very strong consistency guarantees: messages are delivered reliably, in the order they were sent, perhaps with real-time properties. eBay, at least, views every one of these as a dangerous proposition that should be confined, as much as possible, to limited numbers of mission critical subsystems!

3.3.3 Commercial Trends: Virtual Reality (VR), Simulation and Gaming

We also looked at the trends in the VR, simulation and gaming industry. This is a tremendously active community in which new standards, protocols and technologies are emerging at a frantic pace, and are achieving quick market uptake. Huge sums of money are being earned, and spent, and we believe that the long term future of DMO is bright in part because of the potential to leverage these commercial products and technologies.

The steady increase in power and flexibility of technologies such as JavaScript should also be seen as a trend with implications for DMO. With JavaScript and AJAX (asynchronous JavaScript and XML) one can, today, send a personalized advertisement to a user: a talking chameleon with knowledge of that user's insurance situation and a special offer crafted just for her. Tomorrow, these will be powerful enough to represent a missile: one could imagine sending a JavaScript model of a weapon to its target, complete with an accurate model of the way that the weapon steers itself, handles chaff and other anti-weapon technologies, etc. But this is a technical area very much in flux, and it will be some time before such options can be exploited easily.

Thus, in the short term, we see many reasons that DMO will be unable to easily exploit future commercial products – even those just now reaching the market or likely to do so soon. A further concern is that we found that the largest and most credible VR and gaming companies are showing little interest in developing strong military ties. For example, World of Warcraft seems like an obvious match for the DMO community, but in fact the vendor marketing this product is quite explicit in its disinterest relative to military opportunities. Although there are public domain, open source, World of Warcraft servers and client systems, adapting them to the DMO application would pose big challenges.

We believe the situation will remain fluid. Microsoft, one big player in the gaming market, has long maintained dialog with the military and could enter as a possible DMO technology source. Products such as the Xbox would then be plausible technology components for DMO, and the associated wide-area multiuser gaming infrastructure could be potentially adapted for DMO use. Google is well known to be moving to compete with Microsoft in most arenas, and Google Earth seems like an obvious basis for a future VR immersion system which could be of great interest to the DMO community; moreover, unlike the other companies mentioned, Google has an active history of dialog with the government and military (although on the negative side: not just in the United States!)

Concerns that would need to be addressed include the security needs of DMO systems. Because DMO links real-world intelligence databases, satellite imaging systems, DoD maps, and information about real weapons systems, much of the data employed is sensitive and some is classified. Commercial gaming systems lack the mechanisms needed to protect data of this nature, although the features used to prevent users from cheating do represent a reasonable starting point for better security. One would also need to ask whether companies that do much of their development through foreign outsourcing are appropriate DMO technology providers, and to reflect on the implications of the close ties some have established with foreign governments, such as those between Google and the Chinese government and intelligence services.

We conclude that over a longer term, the DMO community is nearly certain to find itself in dialog with major companies, such as Microsoft and Google, and very likely to find powerful, exciting, and extremely scalable solutions in this manner. But we also conclude that for the short term, these are not likely to represent viable options for addressing the scalability challenges posed in the prior section. Moreover, before such steps are taken, the security implications of doing so would need considerable study.

3.3.4 Commercial Trends: WAN Accelerators

We commented that TCP, the transport protocol used in many RTIs to support the HLA publish-subscribe model, collapses dramatically in WAN settings subject either to unusually high latency-throughput product (e.g. fast links to remote destinations), or when there is even a tiny level of packet loss on the WAN link.

A number of companies are developing products aimed at this issue, and the largest companies (e.g. Cisco) are showing interest in the issue. Below, we discuss this in connection with Maelstrom, a Cornell research effort that has achieved striking success in overcoming the underlying problem.

We conclude that WAN network accelerators that function by shielding TCP from the problems seen in WAN deployments are a likely option for deployment in the DMO space. Indeed, there seem to be a great variety of possible options. On the other hand, none of these options is widely accepted and hence none is likely to be used in early DMO systems on the GIG. Lacking them, HLA will very likely exhibit complete performance collapses for at least some patterns of use.

3.3.5 Commercial Trends: WAN Support for UDP Multicast

Because DIS, and some HLA RTI offerings, are multicast based, we looked at the trends in commercial support for UDP multicast over WAN networks. Historically, this has been a problem area: studies dating to one by Wong and Katz at Berkeley in the late 1990s showed that WAN routers degrade dramatically when UDP multicast or broadcast traffic is routed through them, and also that network interface cards (NICs) on inexpensive computers perform poorly when UDP multicast is used aggressively. At the time of this writing, the issue remains quite real: the military GIG will not be a good technology base for wide area UDP multicast.

There is, however, considerable commercial pressure to change the situation. Growth in the popularity of web-based streaming media (videos and advertising) is creating a new market around the delivery of multicast data streams, and hardware is finally advancing to embrace the opportunity. Our study identified a wide range of cutting edge routers with multicast capabilities that, at least if the literature is to be believed, will support large numbers of WAN UDP streams without suffering performance collapse.

We conclude that while the performance instability issues cited earlier are a real concern that will need to be addressed, presumably using better protocols from vendors such as IBM and Red Hat that are investing to solve the problem, the potential does exist for WAN solutions to be developed that could overcome existing limitations and operate effectively in WAN deployments. The core insight here is that investment will be required. Given a space in which existing products are not working well under demanding deployments, core protocol research will be key to developing solutions that scale adequately and are immune from disruptive meltdowns such as the one that was discussed earlier. DoD and the DMO community cannot assume that IBM and Red Hat will solve the problem as encountered in DMO scenarios; it would be very nice if they do so, but their motivations are different and one can imagine a successful IBM eventing product that would not be useful in a WAN DIS deployment simply because of assumptions that hold for IBM's target product area (very likely Web Services systems running Websphere) and do not hold in the DIS setting.

This perspective leads us towards a DMO research recommendation, connected to the topic of Event Nets mentioned earlier.

3.3.6 Research: Real Time and Time-Critical Protocols

Our study reviewed research focused on a problem mentioned earlier in connection with scalability of HLA: when a publish-subscribe system is implemented over point to point UDP or TCP protocols, as in most HLA RTIs, timing skew can become a serious issue. For example, if we imagine a popular publisher monitored by 5000 entities, 5000 TCP sends may elapse between when the first entity learns of an event and the last one. Suppose that the event in question is a missile launch. That 5000th receiver may not learn of the event until many seconds after the first learned of it.

AFRL has long been aware of this problem and has a history of investing to develop solutions. The AFRL Castor project conducted at Cornell developed a powerful new protocol, Ricochet, that addresses the issue and achieves dramatic latency improvements (many orders of magnitude) and seems like a very good match with applications that use publish-subscribe communication patterns. Indeed, even DIS applications could benefit.

The Cornell protocol, called Ricochet, exists as a public-domain, open source distribution. Dialog is underway between Cornell and Red Hat, the main Linux vendor, to integrate Ricochet into the Red Hat QPID product, targeted for 2010. (We refer the reader to the published Ricochet paper for details.)

3.3.7 Research: WAN Network Protocols

AFRL has also supported research into WAN communication solutions aimed at eliminating the TCP performance collapse noted earlier, and identified as a serious threat to HLA deployments in WAN settings. Maelstrom, a research solution developed with AFRL funding, addresses this issue by introducing a performance accelerating proxy that resides on each end of a WAN network connection, for example at two DMO sites that are connected using HLA. Maelstrom injects coded redundancy using a novel form of forward error correction and takes other actions, such as aggregating flows and terminating them locally, to transparently accelerate TCP and to mask the impact of latency and packet loss. As noted earlier, there are also commercial products in this space, but Maelstrom appears to be far more transparent and effective than any reported product on the market, or announced for the future. (We refer the reader to the published Maelstrom paper for details.)

3.3.8 Research: Distributed System Monitoring

The mechanisms proposed here often require some degree of distributed management or monitoring. For example, it may be necessary to collect status information from the endpoint applications so as to configure the WAN tools correctly, or to monitor for and detect evidence of instability that might then be used to justify a decision to disable the use of a protocol such as UDP multicast in favor of a slower but more stable alternative. In a number of AFRL and DARPA funded efforts, Cornell has worked on this monitoring problem and developed a number of solutions, some of which were eventually commercialized. For example, our Astrolabe system was ultimately acquired by Amazon and deployed in their data centers for system self-management.

The key Cornell technology in the monitoring area involves what is called peer-to-peer or gossip communication, a particularly scalable way of sharing information acquired in all-to-all communication settings. We will not repeat content from the published Astrolabe paper, but we do note that similar mechanisms could be created for the DMO community and used to automate configuration management and adaptation.

4. Transparent Intervention

Our review makes it clear that there are a number of plausible technologies for use in DMO and that, if used appropriately, they offer responses to at least some of the worrying problems DMO is likely to encounter in the absence of careful planning. But they also pose a serious question: how can these solutions be adopted in light of the legacy constraint cited earlier? After all, if DMO cannot be changed, what choice do we have other than to blindly push forward?

AFRL previously identified this issue and asked Cornell to look at the question as part of the Castor project. Cornell's Dr. Multicast effort reveals a promising technical option.

The basic idea of Dr. Multicast dates back to one of the tricks used by hackers when breaking into a Linux or Windows computer. Modern operating systems generally share library procedures between applications. For example, suppose a set of applications all use the system provided "Write" system call, or the "printf" formatted IO interface.

Rather than loading a copy of the needed procedure for each application that uses it, a single copy is loaded as part of a shared library, and then the running applications are linked to the library.

The actual mechanism is fairly simple, although it involves some trickery we will not review here. But basically, when the application is compiled, a stub is created that will track down the library at runtime by looking for a suitable file in a search path: a sequence of places that will be checked, in order. The first match is dynamically linked to the application that uses it, hence we call the associated file a "dynamically linked library" or DLL.

The hackers had the idea of creating a non-standard library and placing it somewhere on the search path. Thus, the application "thinks" it is linking itself to the standard printf routine, but actually finds itself linked to the hacked one. That routine might pretend to be printf in all visible ways, but quietly do something else, such as sending a copy of all the output to a machine accessed over the Internet.

Our approach, in Dr. Multicast, is very similar, but for a good rather than a nefarious purpose. We create a library that exposes some standard interface, like the UDP broadcast interface used by DIS. Now, when an application uses DIS, we will get a chance to intercept the operations it performs (Figure 5, below, and Figure 6). This gives us a chance to replace the standard UDP broadcast with something else. We can substitute Ricochet, or route the messages through IBM's cutting-edge Gravity protocol, or even use the TIBCO product.

Moreover, through a bit of trickery, the interposition library can still access the standard one. Just as a hacker might intercept printf calls, make a copy of the data, and then invoke the standard printf, we can intercept calls to DIS, inspect them, and if we wish, hand them off to a standard DIS implementation.

The possibility then exists of building a filtering function that will let the applications continue to think of themselves as doing DIS eventing, with its broadcast semantics, and yet substitute a publish-subscribe functionality, in which only desired traffic ever reaches receivers. True, we would need a way to understand what packet the receiver is filtering and which will get through, but if a product exposes some API by which it learns to filter, we can potentially hijack that API to obtain this filtering information in our system.

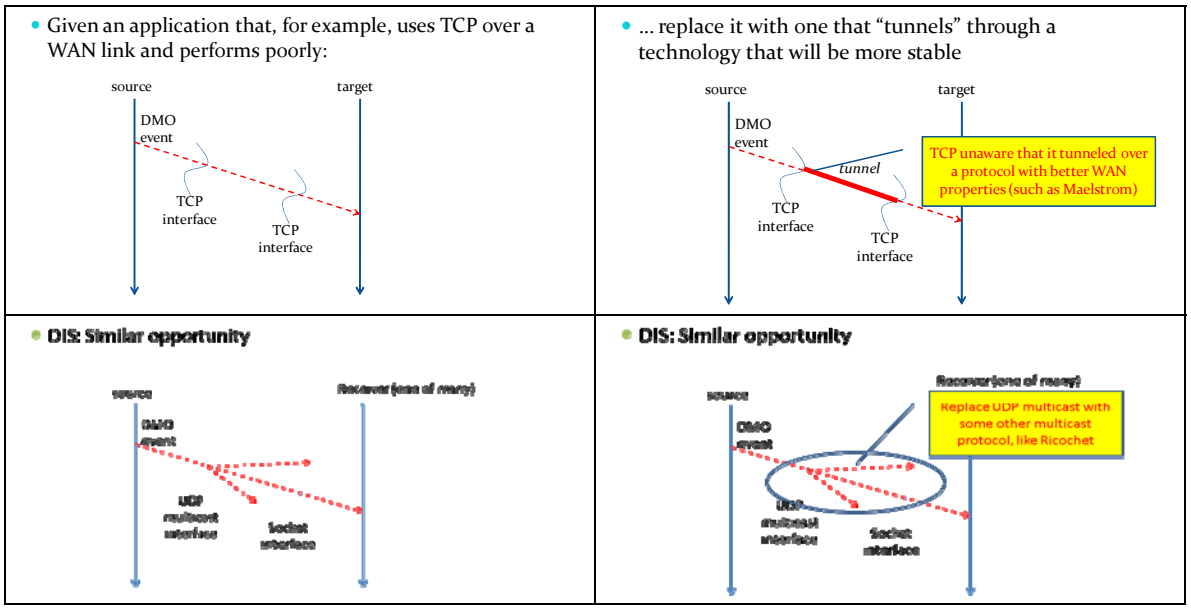

Figure 5: "Under-the-Hood" Transparent Intervention

HLA offers a similar opportunity. One possibility is to intercept TCP operations. But HLA itself has a standard API, and we can potentially intercept HLA requests at that level as well. The same can potentially be done for TENA, and for DMON.

The appeal of this approach is its extreme transparency. Just as Maelstrom hides beneath TCP and fixes a problem that would disable TCP, Dr. Multicast hides under a variety of standard APIs and fixes problems that can cause instability when those APIs are employed directly and without changes. In theory, no change to the application is required at all, hence we can usually work with a compiled application for which code is not available.

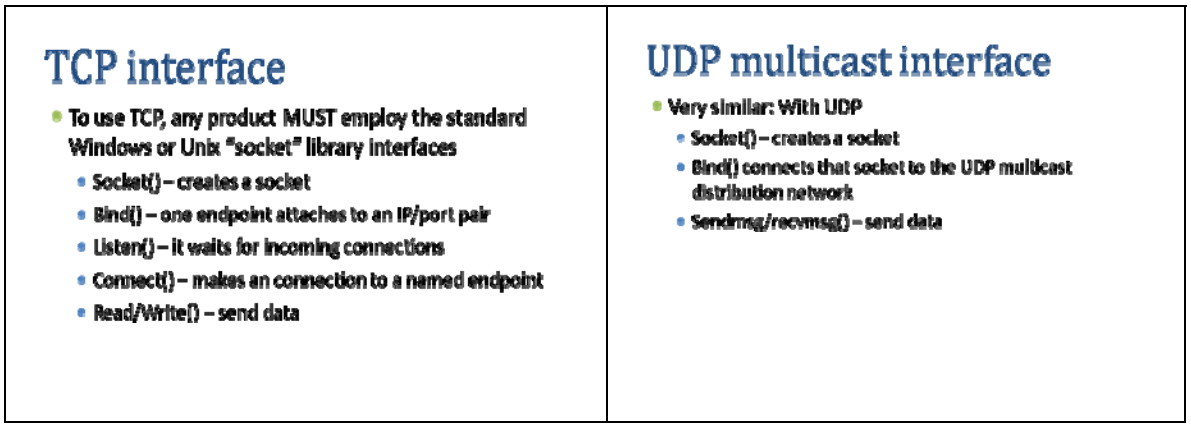

Figure 6: TCP and UDP are well suited for Transparent Intervention

We refer the reader to the published Dr. Multicast paper for details.

5. Recommendations

We conclude this report with a summary of our recommendations. These fit broadly into a two-pronged strategy that invests first in short-term solutions aimed at fixing obvious scalability problems as transparently as possible, and long term study aimed at clarifying options so that as DMO advances, we will know how best to exploit the technical possibilities that are likely to emerge or that can be created through research.

5.1 Near term steps

Our core recommendations are as follows.

1. DMO should develop a toolkit using ideas similar to those in Dr. Multicast to enable insertion of new protocols underneath DIS and HLA in existing applications that cannot easily be modified and for which source may be unavailable.
2. Working with vendors, DMO should exploit the new leverage created by these intervention tools to deploy best-of-breed alternatives that would replace the normal communication technologies employed in DIS implementations and HLA RTIs (and similarly for TENA and DMON, down the road). We have pointed to some of the options, such as the future Red Hat version of Ricochet or IBM's future distributed computing system. Others may not have been identified in this study, but this does not preclude their use. In Section 3.3.1 we pointed out that there are no commercial studies that compare technology options side-by-side. The DMO community may need to undertake experiments in order to select best options. This is a slow and difficult process and can be costly, hence any pre-selection that can be performed is highly desirable.
3. HLA versions that run over TCP could be mapped to use Maelstrom in WAN deployments, or one of the commercial data center linkage products mentioned above.
4. DMO should develop a distributed system monitoring and management solution, perhaps using ideas like those in Astrolabe, to increase the degree of control and oversight that can be performed when administering large WAN deployments. Doing so will head off melt-downs such as the ones described in Section 3.3.1.

We estimated risk and reward as an aspect of the RAPID study and concluded that these short-term steps have limited risk. Potential risks include:

1. Although transparent intervention into UDP broadcast is definitely feasible, the same technique may be harder to adapt for use with HLA, TENA and DMON.
2. It may be difficult to extract DIS filtering information from unmodified receiver applications, yet this is key to reducing the overall system load.
3. Ricochet, the DIS replacement identified above, is currently still under consideration by Red Hat; that company could ultimately decide not to proceed with the project or might adopt some other, less suitable solution.
4. DIS applications may be so sensitive to features of the existing solutions that they would malfunction if anything is changed in any way at all.
5. Data center linkage tools like Maelstrom may turn out to be incompatible with WAN security mechanisms needed in GIG settings. The small companies

mentioned may fail, and Maelstrom, the Cornell technology, may never be commercially supported.

6. DMO vendors may resist the proposed approaches, or favor alternatives.

Of these risks, we think that the last is perhaps the most likely. But we also believe that if AFRL can demonstrate the potential for great improvements in DMO scalability and performance, a vendor advocating a different approach would have some obligation to do "even better". Thus it may be feasible to elevate the dialog and raise the technical bar. Even if the specific DMO interventions we recommend are never adopted by the DMO community, once vendors see a team demonstrate these sorts of options, the idea of using them cannot easily be removed from the table, and the odds are that the vendors will simply recommend their own approaches, and argue for the superiority of their alternatives. Success breeds imitation, and in this case, imitation would be a successful outcome!

The rewards associated with the steps we recommend could be substantial. If our study is correct, absent some form of mid-course correction, DMO faces a very challenging period. The steps recommended here could be key to breaking through those technology barriers and taking DMO to the next level of scalability.

5.2 Long term steps

We conclude by recommending a plan of longer term research and investment aimed at creating the options DMO will need in the future, as it advances beyond today's technology limits and challenges.

First, we strongly recommend that DMO gain experience using JavaScript and AJAX. These are powerful tools for creating, in effect, a new kind of mobile "event" that could open the door to dramatic new DMO functionalities. More broadly, the DMO community needs to start looking at technologies such as Google Earth, Second Life and World of Warcraft and to begin positioning itself for a future in which those will become so powerful and so pervasive that working with them becomes an inevitable step. Gateways need to be developed and thought invested in the security implications of such a future.

Second, we urge DMO and AFRL to invest in a new area that we term an "Event Net". The National Science Foundation recently created a new program of research in the area of Data Nets, which it defines as follows: *Science and engineering research and education are increasingly digital and increasingly data-intensive. Digital data are not only the output of research but provide input to new hypotheses, enabling new scientific insights and driving innovation. Therein lies one of the major challenges of this scientific generation: how to develop the new methods, management structures and technologies to manage the diversity, size, and complexity of current and future data sets and data streams. [The NSF] solicitation addresses that challenge by creating a set of exemplar national and global data research infrastructure organizations (dubbed DataNet Partners) that provide unique opportunities to communities of researchers to advance science and/or engineering research and learning.*

The new types of organizations envisioned in this solicitation will integrate library and archival sciences, cyberinfrastructure, computer and information sciences, and domain science expertise to:

- *provide reliable digital preservation, access, integration, and analysis capabilities for science and/or engineering data over a decades-long timeline;*
- *continuously anticipate and adapt to changes in technologies and in user needs and expectations;*
- *engage at the frontiers of computer and information science and cyberinfrastructure with research and development to drive the leading edge forward; and*
- *serve as component elements of an interoperable data preservation and access network.*

By demonstrating feasibility, identifying best practices, establishing viable models for long term technical and economic sustainability, and incorporating frontier research, these exemplar organizations can serve as the basis for rational investment in digital preservation and access by diverse sectors of society at the local, regional, national, and international levels, paving the way for a robust and resilient national and global digital data framework.

It is interesting to note that DARPA is also working to create a similar program focused on DoD biometric databases and associated challenges. The DARPA program is still at a discussion stage, but we see the mere fact that a discussion is occurring as evidence supporting our contention that the time for DataNet and EventNet research has arrived.

Our finding is that the DMO challenge has strong similarities to the two programs just summarized. Clearly, DMO is not focused on infrastructure for future scientific research, and needs to support a much more event-driven "flavor" of application. However, one can also see the connections: DMO, like a Data Net, will draw on huge banks of precomputed data and scenarios, maps, intelligence, video and other sensor information.

Indeed, we see DMO as a kind of Event Net: a Data Net augmented by massive event streams carrying huge rates of transient updates that dovetail with the archival data residing in the underlying Data Net storage systems.

Accordingly, we recommend that the DMO and AFRL explore the creation of an Event Net research program aimed at creating capabilities to support ambitious, globally deployed Event Nets that could be used in defense simulation, modeling, training, and perhaps also in actual mission planning and execution. The potential appears to be enormous and the timing very appropriate.

References

- Charles.Cashulette. "DMO Focus Study." Air Force Research Laboratory, Information Directorate, Systems and Information Interoperability Branch. July 2008.

- Ken Goad. “DoD LVC Architecture Roadmap (LVCAR) Study Status,” March 2008.
- Amy Henninger, Bob Richbourg, Bob Lutz, Brian Williams, and Margaret Loper. “Live Virtual Constructive Architecture Roadmap (LVCAR) Interim Report.” March 2008.
- Margaret Loper and Dannie Cutts. “Live Virtual Constructive Architecture Roadmap (LVCAR) Comparative Analysis of Standards Management.” March 2008.
- Robert Richbourg and Robert Lutz. “Live Virtual Constructive Architecture Roadmap (LVCAR) Comparative Analysis of the Middlewares.”
- Ke Pan, Stephen John Turner, Wentong Cai, and Zengxiang Li. “A Service Oriented HLA RTI on the Grid.” In Proc. of IEEE International Conference on Web Services (ICWS), July 2007.
- Mahesh Balakrishnan, Ken Birman, Amar Phanishayee, and Stefan Pleisch. “Ricochet: Lateral error correction for timecritical multicast.” In Proc. of USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2007.
- Mahesh Balakrishnan, Tudor Marian, Ken Birman, Hakim Weatherspoon, and EinarVollset. “Maelstrom: Transparent error correction for lambda networks.” In Proc. of USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2008.
- Ymir Vigfusson, Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, and Yoav Tock. “Dr. Multicast: Rx for Datacenter Communication Scalability.” In Proc. of ACM Workshop on Hot Topics in Networks (HotNets), October 2008.

HLA RTI Implementations

Commercial HLA RTI Vendors

- Chronos RTI. Magnetar Games. http://www.magnetargames.com/Products/Chronos
- MaK High Performance RTI. MaK Technologies. http://www.mak.com/products/rti.php
- HLA Direct. General Dynamics. http://www.gdc4s.com/documents/S2FocusWhitePaper.pdf
- Openskies RTI. Cybernet Systems. http://www.openskies.net
- Pitch pRTI. Pitch Technologies. http://www.pitch.se/products/pitch-prti/pitch-prti-overview.html
- RTI NG Pro. Raytheon Virtual Technology Corporation. http://www.raytheonvtc.com/products.jsp
- Raytheon Virtual Technology Corporation & SAIC - RTI NG Pro. http://www.virtc.com/rtingpro-full.jsp
- Aegis/PitchAB (Sweden). http://www.aegistg.com/

Non-commercial HLA RTI Vendors

- BH-RTI. Beijing University of Aeronautics and Astronautics Virtual Reality Laboratory. http://www.hlarti.com, http://translate.google.com/translate?u=http%3A%2F%2Fwww.hlarti.com%2F&hl=en&ie=UTF8&sl=zh-CN&tl=en

- CERTI. ONERA. http://savannah.nongnu.org/projects/certi
- EODiSP HLA. P&P Software. http://www.pnp-software.com/eodisp
- GERTICO (German RTI based on Corba). Fraunhofer IITB. http://www.iitb.fraunhofer.de/servlet/is/2920
- The Portico Project. Littlebluefrog labs. http://porticoproject.org
- Open HLA. http://sourceforge.net/projects/ohla
- RTI-S. US JFCOM J9 Directorate. http://dss.ll.mit.edu/dss.web/stow.html

Acronyms

API	Application Program Interface (e.g. to a software library)
DARPA	US Defense Advanced Research Projects Agency
DDL	Dynamically linked library
DIS	Distributed Interactive Simulation
DMO	Distributed Mission Operations
DMON	Distributed Mission Operations Network
DMSO	Defense Modeling and Simulation Office
DSSA	TENA Domain-Specific Software Architecture
FED	Federation Execution Data
FOM	Federation Object Model
GIG	Global Information Grid
HLA	High Level Architecture
IA	Information Assurance
LAN	Local Area Network
MOM	HLA Management Object Model
RTI	HLA Run Time Infrastructures
TENA	Training Enabling Architecture
WAN	Wide Area Network
WS	Web Services, the technology underlying the GIG standards

Terms With Special Meanings

Cloud Computing	A style of computing that offloads tasks to large data centers
Virtual Machine	A way of using one computer to "host" another computer, so that the hosted system runs on the host system. The hosted image seems just like a standard one, but can be moved about to balance loads

Terms Used For Communication Protocols

TCP	Internet "data streaming" protocol: for sending streams of data reliably
UDP	Internet "packet" protocol: supports point-to-point data messages with no reliability or ordering guarantees
Broadcast	A mode in which a UDP message will be delivered to *every application using the DIS platform.* Reliability is not guaranteed, so data can be lost

Multicast	A mode in which a UDP message is delivered to a *set of applications* that register to listen to the multicast *address*. Reliability is not guaranteed.
Ordered	The guarantee that data will arrive in the order it was sent. TCP provides this property, but UDP does not.
Reliable	The guarantee that no data will be lost, even if a network packet must be resent. Requires some kind of dialog between the sender and receiver
Publish-Subscribe	An API allowing programs to "publish" information labeled by "topic" and permitting others to "subscribe" by topic. Can be implemented by a set of TCP connections, or using UDP multicast
Filtering	A term used when a program may receive messages it did not actually request as a side-effect of using broadcast. Such messages must be "filtered out" and ignored or they could crash the receiving program.
Real Time	A protocol that guarantees speedy delivery of messages